

An Improved Method for Numerical Conformal Mapping

By John K. Hayes, David K. Kahaner and Richard G. Kellner*

Abstract. A new technique for the numerical conformal mapping of a planar region onto the unit disk has been presented and tested by Symm. By elaborating on his methods, we have improved the accuracy of the numerical results by up to four orders of magnitude. For illustration, our methods have been applied to several of the same regions considered in the literature by Symm and Rabinowitz. A flexible FORTRAN code and User's Guide are reproduced on the microfiche card in this issue.

1. Introduction. A new technique for the numerical conformal mapping of a planar region onto the unit disk has been presented and tested by Symm [7], [8], [9]. By elaborating on his methods, we have improved the accuracy of the numerical results by up to four orders of magnitude. For illustration, our methods have been applied to several of the same regions considered in the literature by Symm [7] and Rabinowitz [6].

In this paper, we numerically approximate the univalent function $f(z)$ which maps the bounded, simply-connected region D of the complex plane onto the unit disk. Let L be the boundary of D and choose $z_0 \in D$ to be the point which is to be mapped into the center of the unit disk. It is known [7] that

$$w = f(z) = \exp[\log(z - z_0) + g(z) + ih(z)],$$

where g and h are real-valued harmonic conjugates, and g satisfies

$$\nabla^2 g(z) = 0 \quad \text{for } z \in D,$$

and

$$g(z) = -\log |z - z_0| \quad \text{for } z \in L.$$

The mapping function $f(z)$ above is determined only to within an arbitrary rotation. This depends upon the branch of the logarithm used in the computation and the additive constant chosen for the function h .

Symm [7] numerically solves the integral equation of the first kind

$$(1) \quad \int_L \sigma(\zeta) \log |z - \zeta| |d\zeta| = -\log |z - z_0|, \quad z \in L.$$

This may always be done, subject to a possible rescaling of the region D [3], [5]. Then, for any $z \in D + L$, g and h have the representation

Received June 21, 1971.

AMS 1970 subject classifications. Primary 30A28; Secondary 65E05.

Key words and phrases. Numerical conformal mapping, numerical solution of integral equations of the first kind.

* This research was supported by the U. S. Atomic Energy Commission under Contract W-7405-ENG-36.

Copyright © 1972, American Mathematical Society

$$(2) \quad g(z) = \int_L \sigma(\zeta) \log |z - \zeta| |d\zeta|,$$

$$(3) \quad h(z) = \int_L \sigma(\zeta) \arg(z - \zeta) |d\zeta|.$$

The function \arg must be chosen in an appropriate manner [4].

2. Description of the Method. Our procedure for numerically mapping a region can be divided into two operational steps:

(i) Solve Eq. (1) for the function σ .

(ii) Evaluate Eqs. (2) and (3) for each point $z \in D$ where we want to find $f(z)$.

Let the curve L have the parametric representation $\{(\nu(t), w(t)) \mid t \in (0, d]\}$ with respect to arc length t . Here, d is the length of L . Define $\zeta(t) = \nu(t) + iw(t)$. With this notation, Eqs. (2) and (3) become

$$(4) \quad g(z) = \int_0^d \sigma(t) \log |z - \zeta(t)| dt, \quad z \in D + L,$$

$$(5) \quad h(z) = \int_0^d \sigma(t) \arg(z - \zeta(t)) dt, \quad z \in D + L,$$

where we have used $\sigma(t)$ for $\sigma(\zeta(t))$.

Now, we will sketch how we compute the function $\sigma(t)$ numerically. A detailed development is contained in [1]. Since $\sigma(t)$ is a function of arc length, we extend it continuously as a periodic function on $(-\infty, +\infty)$. For ease of explanation, assume $\sigma(t) \in C^3(-\infty, +\infty)$ and that L has no corners. Place on L a uniform mesh of n points (n even), each $h = d/n$ units apart. (In actual practice, one might wish to divide L into several sections. The mesh points on each section would then be uniform with respect to arc length on that section. See the user's guide in the microfiche portion of this issue and also Example 2 of this paper.) Define a set of piecewise polynomial functions $p_1(t), p_2(t), \dots, p_n(t)$ by

$$\begin{aligned} p_1(t) &= (t - h)(t - 2h)/2h^2, & 0 \leq t \leq 2h, \\ &= (t + h)(t + 2h)/2h^2, & -2h \leq t \leq 0, \\ &= 0, & \text{otherwise,} \\ p_2(t) &= -t(t - 2h)/h^2, & 0 \leq t \leq 2h, \\ &= 0, & \text{otherwise,} \\ p_{2i+1}(t) &= p_1(t - 2ih), & i = 1, 2, \dots, n/2 - 1, \end{aligned}$$

and

$$p_{2i}(t) = p_2(t - 2(i - 1)h), \quad i = 2, 3, \dots, n/2.$$

Define also $\bar{\sigma}(t) = \sum_{i=1}^n \sigma(ih)p_i(t)$. It is true that

(i) $\bar{\sigma}(t)$ is a polynomial of degree two on $[ih, (i + 2)h]$, for $i = 0, 2, 4, \dots, n - 2$.

(ii) $\bar{\sigma}(t) = \sigma(t)$ at $t = ih$, for $i = 0, 1, 2, \dots, n$.

$$(6) \quad \text{(iii)} \quad \sigma(t) = \bar{\sigma}(t) + O(h^3) = \sum_{i=1}^n \sigma_i p_i(t) + O(h^3),$$

where $\sigma_i = \sigma(ih)$ for $i = 1, 2, \dots, n$.

Using the approximation Eq. (6) in Eq. (4), we get

$$(7) \quad \sum_{k=1}^n \sigma_k \int_0^d p_k(t) \log |z - \zeta(t)| dt = g(z) + O(h^3).$$

The function $g(z) = -\ln |z - z_0|$ for $z \in L$. Thus, we can evaluate Eq. (7) at the points $z = \zeta(ih)$ for $i = 1, 2, \dots, n$, and we will get n linear equations with constant coefficients for the variables $\sigma_1, \sigma_2, \dots, \sigma_n$. Set $A = (a_{ik})$ and $B = (b_i)$, where

$$a_{ik} = \int_0^d p_k(t) \log |\zeta(ih) - \zeta(t)| dt, \quad \text{for } i, k = 1, 2, \dots, n,$$

$$b_i = -\log |\zeta(ih) - z_0|, \quad \text{for } i = 1, 2, \dots, n.$$

With this notation, Eq. (7) leads to the linear system $A\delta = B + O(h^3)$, where $O(h^3)$ is a vector, with each component bounded by $O(h^3)$, and $\delta = (\sigma_1, \sigma_2, \dots, \sigma_n)^T$.

The matrix equation we actually solve is

$$(8) \quad \tilde{A}\tau = B,$$

where the elements of \tilde{A} are approximations to those of A . Using our representation for the $p_k(t)$, it is evident that to compute A it is sufficient to evaluate integrals of the form

$$(9) \quad \int_{(i-1)h}^{ih} t^j \log |z - \zeta(t)| dt$$

for $i = 1, 2, \dots, n$ and $j = 0, 1, 2$. The \tilde{a}_{ij} are the result of approximating the integrals (9). For each fixed x, y and i , we approximate $|z - \zeta(t)|$ by a polynomial $q(t)$ of degree two on $((i-1)h, ih)$. We choose $q(t)$ so that

$$q(t) = |z - \zeta(t)|^2 \quad \text{for } t = (i-1)h, (i-\frac{1}{2})h, ih.$$

Then

$$\int_{(i-1)h}^{ih} t^j \log |z - \zeta(t)| dt \approx \frac{1}{2} \int_{(i-1)h}^{ih} t^j \log[q(t)] dt.$$

The integrals on the right-hand side above can be evaluated explicitly. For certain special cases, for instance when $|z - \zeta(t)| = 0$ on $[(i-1)h, ih]$, the treatment is slightly different in that a higher order polynomial is used.

We then solve the matrix equation $\tilde{A}\tau = B$ for the vector $\tau = (\tau_1, \tau_2, \dots, \tau_n)^T$ and use this as an approximation to δ . Then

$$\|\delta - \tau\| \leq \|(A^{-1} - \tilde{A}^{-1})B\| + \|A^{-1}\| O(h^3).$$

We have found by experience on numerous problems that the error due to $A^{-1} - \tilde{A}^{-1}$ seldom if ever dominates the $\|A^{-1}\|O(h^3)$ term. Another analysis [2] strongly indicates that $\|A^{-1}\| \leq O(1/h)$.

Once $\sigma(t)$ has been computed, we may calculate $g(z)$ and $h(z)$.

$$\begin{aligned} g(z) &= \int_0^d \sigma(t) \log |z - \zeta(t)| dt \\ &\approx \sum_{k=1}^n \tau_k \int_0^d p_k(t) \log |z - \zeta(t)| dt. \end{aligned}$$

These integrals are approximated as described above. The calculation of $h(z)$ is more difficult. Using integration by parts, and approximations similar to those above, we are led to integrals of the form

$$\int_{(i-1)h}^{ih} t^j \arg(z - \zeta(t)) dt,$$

where $i = 1, 2, \dots, n$ and $j = 0, 1, 2, 3$. The evaluation of these integrals is discussed in detail in [1].

The method set forth by Symm in [7] uses piecewise constant functions in Eq. (6) and evaluates the integrals in Eq. (7) by using Simpson's rule of integration.

3. Tests. A FORTRAN IV version of the algorithm described has been coded to run on our CDC 6600 and CDC 7600. This program is more or less machine independent, has flexible input, and is general enough to handle a large class of problems. It is a modification of a program described in [1] which has been in use for a few years. A limited number of copies of this deck and a user's guide are available from the authors. Using this code, we have computed some approximate conformal maps for several regions, including some used in [7]. Since our technique is an extension of the method used there, it is appropriate to compare our results with those. All of the regions selected for test have substantial symmetry. We have elected to ignore this symmetry in our code in order to give utmost flexibility. Taking advantage of symmetry ought to enhance the accuracy by reducing the volume of computation.

Symm has pointed out [8] that the maximum errors occur on the boundary of the region being mapped. Since points on the boundary have image points on the unit circle, it is easy to check the error in the modulus of an arbitrary boundary point. The data points themselves are constrained by the defining equations to be mapped onto $|w| = 1$, hence we check for modulus error at points midway between each of the data points. The columns labeled ERR-MOD contain the maxima of the quantities $||w| - 1|$ at these intermediate points. Computing the error in the argument is more difficult. Symm provides an estimate of this in [8], denoted E_A . Our experience has indicated that as the region becomes less circular and more elongated, errors, particularly those in the argument, increase in a monotonic way. Since the numbers E_A provided by Symm did not have this property, we considered them somewhat unreliable and decided to use an alternative technique. The columns labeled ERR-ARG represent the maximum difference in the argument at the data points between two computations, the second corresponding to the largest number of data points used for the domain in question. It is reasonable to examine the argument at the data points rather than the intermediate points, since the argument is not constrained in any way by Eq. (1). This procedure does yield the monotonicity we expect. In certain cases, analytic expressions for the conformal maps are available. It is then possible to compute the absolute errors in the argument exactly. These numbers compare extremely well with the approximate errors ERR-ARG described above, and constitute our main justification for this approach.

Each of our test regions has its center point mapped into the origin. In what follows, h and n will have the same meaning as in Section 2.

Example 1. Oval of Cassini. This curve is defined by

$$[(x+1)^2 + y^2][(x-1)^2 + y^2] = \alpha^4, \quad \alpha > 1.$$

For α near 1, the curve is elongated and nonconvex, becoming more circular as α increases. For $\alpha = 1.06$, the width to height ratio is about 5. Points are distributed uniformly on the entire boundary. The exact mapping is given by

$$f(z) = \alpha z / (\alpha^4 - 1 + z^2)^{1/2},$$

and we use this to compute errors in the argument.

TABLE I
Oval of Cassini

α	n	h	ERR-MOD	ERR-ARG
1.06	65	.11	2×10^{-3}	2×10^{-3}
1.06	129	.06	1×10^{-4}	2×10^{-4}
1.8	33	.34	1×10^{-4}	1×10^{-4}
1.8	65	.18	1×10^{-5}	1×10^{-5}
1.8	129	.09	6×10^{-7}	7×10^{-7}

The maximum error occurs at or near $x = 0$. The errors near $y = 0$ are smaller by a factor of 1/100. The comparison with Symm must be made carefully, since his data points are for the most part distributed uniformly with respect to x rather than t . As far as we can determine, errors in the modulus are from one to four orders of magnitude better than those in [7]. We should emphasize that our distribution of points is poor.

For $\alpha = 1.06$, $n = 65$, Table II indicates errors for points inside the curve.

TABLE II
Oval of Cassini

$\alpha = 1.06$	$n = 65$	ERRORS IN	
x	y	MODULUS	ARGUMENT
1.4	0.	6×10^{-7}	3×10^{-5}
1.26	0.	6×10^{-7}	3×10^{-5}
1.12	0.	1×10^{-6}	3×10^{-5}
0.98	0.	2×10^{-6}	3×10^{-5}
0.84	0.	2×10^{-6}	3×10^{-5}
0.7	0.	4×10^{-6}	3×10^{-5}
0.56	0.	7×10^{-6}	3×10^{-5}
0.42	0.	2×10^{-5}	4×10^{-5}
0.28	0.	5×10^{-5}	4×10^{-5}
0.14	0.	1×10^{-4}	4×10^{-5}

Example 2. Rectangle. $-1 \leq x \leq +1, -\alpha \leq y \leq \alpha$.
The case $\alpha = 1$ was computed exactly by the use of elliptic integrals. In the other cases, we use a comparison with the most accurate computed values. Points are uniformly spaced on each side, with $n/4$ points per side. See Table III.

TABLE III
Rectangle

α	n	ERR-MOD	ERR-ARG
0.1	516	4×10^{-5}	—
	260	6×10^{-4}	7×10^{-4}
	132	5×10^{-3}	6×10^{-3}
	68	1×10^{-2}	1×10^{-2}
	36	6×10^{-2}	5×10^{-2}
0.2	516	3×10^{-6}	—
	260	4×10^{-5}	4×10^{-5}
	132	5×10^{-4}	6×10^{-4}
	68	5×10^{-3}	7×10^{-3}
	36	1×10^{-2}	2×10^{-2}
0.4	260	3×10^{-6}	—
	132	4×10^{-5}	4×10^{-5}
	68	6×10^{-4}	7×10^{-4}
	36	5×10^{-3}	1×10^{-2}
0.5	260	1×10^{-6}	—
	132	2×10^{-5}	2×10^{-5}
	68	2×10^{-4}	2×10^{-4}
	36	2×10^{-3}	3×10^{-3}
0.8	260	2×10^{-7}	—
	132	3×10^{-6}	5×10^{-6}
	68	4×10^{-5}	8×10^{-5}
	36	6×10^{-4}	3×10^{-3}
1.0	260	9×10^{-8}	—
	132	1×10^{-6}	1×10^{-6}
	68	2×10^{-5}	3×10^{-5}
	36	2×10^{-4}	1×10^{-3}

Both ERR-MOD and ERR-ARG are monotonic with respect to n and α for $\alpha \leq 1$. Errors in the modulus are from one to two orders of magnitude better than in [7]. It should be noted that for small α the distribution of boundary points is poor. This is true for most of the examples. The only reason for using the given distribution of points is to compare with [7]. Our experience shows that a good rule of thumb

for the distribution of boundary points is to keep the distance between successive boundary points and the distance from the boundary points to the center in a nearly constant ratio. Thus, for α small, we want more points near the centers of the longer two sides and fewer points on the shorter two sides. This can be done by dividing the boundary into sections as mentioned in the paragraph following Eq. (5). We ran the problem with $\alpha = 0.1$ again, using a particularly simple redistribution of the boundary points. For a fixed number of points, the errors decreased by about 1/50. Using an optimal distribution of points, one would get more accuracy.

Example 3. Ellipse. $x^2/\alpha^2 + y^2 = 1$. The data points are uniformly distributed on the boundary of the ellipse. See Table IV.

TABLE IV

Ellipse

α	n	ERR-MOD	ERR-ARG
1.25	257	3×10^{-8}	—
	129	3×10^{-7}	2×10^{-7}
	65	4×10^{-6}	4×10^{-6}
	33	5×10^{-5}	4×10^{-5}
2.5	257	3×10^{-7}	—
	129	4×10^{-6}	5×10^{-6}
	65	5×10^{-5}	6×10^{-5}
	33	7×10^{-4}	9×10^{-4}
5.0	257	4×10^{-6}	—
	129	4×10^{-5}	5×10^{-5}
	65	7×10^{-4}	2×10^{-3}
	33	6×10^{-3}	5×10^{-3}
10	257	4×10^{-5}	—
	129	6×10^{-4}	6×10^{-4}
	65	5×10^{-3}	6×10^{-3}
	33	1×10^{-2}	6×10^{-2}
20	257	5×10^{-4}	—
	129	5×10^{-3}	6×10^{-3}
	65	1×10^{-2}	3×10^{-3}
	33	7×10^{-2}	5×10^{-2}

Again, we note monotonicity with respect to n and α for $\alpha \geq 1$, with maximum error near the center of the side intersected by the minor axis. Improvements over [7] are from one to three orders of magnitude, with the least improvement for $\alpha = 20$.

Example 4. Isosceles Triangle. The corners of the triangle are at $(0, 1)$, $(2, -1)$, $(-2, -1)$, and $(0, 0)$ is mapped into the origin of the unit circle. There are equal numbers of points on each side.

TABLE V
Triangle

<i>n</i>	ERR-MOD	ERR-ARG
65	1×10^{-6}	—
33	2×10^{-5}	6×10^{-5}
17	2×10^{-4}	9×10^{-4}

4. **Timing.** There are three operations that are important as far as timing is concerned: (i) generating the matrix \tilde{A} of Eq. (8), (ii) solving the matrix Eq. (8), and (iii) evaluating the function $f(z)$ at a given point. The time required for (i) is proportional to n^2 and is about 0.85 sec** for $n = 200$. The time required for (ii) is proportional to n^3 and is about 2 sec for $n = 200$. The time required for (iii) is proportional to n and is 0.016 sec for $n = 200$.

University of California
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87544

1. J. HAYES, *Four Computer Programs Using Green's Third Formula to Numerically Solve Laplace's Equation in Inhomogeneous Media*, Los Alamos Scientific Laboratory Report, LA-4423, April 1970.
2. J. HAYES & R. KELLNER, *The Eigenvalue Problem for a Pair of Coupled Integral Equations Arising in the Numerical Solution of Laplace's Equation*, Los Alamos Scientific Laboratory Report, LA-DC-12009, October 1970.
3. M. A. JASWON, "Integral equation methods in potential theory. I," *Proc. Roy. Soc. Ser. A*, v. 275, 1963, pp. 23-32. MR 27 #4034.
4. M. MAITI, "A note on the integral equation methods in potential theory," *Quart. Appl. Math.*, v. 25, 1968, pp. 480-484. MR 37 #447.
5. N. I. MUSKHELIŠVILI, *Singular Integral Equations. Boundary Problems of Function Theory and their Application to Mathematical Physics*, OGIZ, Moscow, 1946; English transl., Noordhoff, Groningen, 1953. MR 8, 586; MR 15, 434.
6. PHILIP RABINOWITZ, "Numerical experiments in conformal mapping by the method of orthonormal polynomials," *J. Assoc. Comput. Mach.*, v. 13, 1966, pp. 296-303.
7. G. T. SYMM, "An integral equation method in conformal mapping," *Numer. Math.*, v. 9, 1966, pp. 250-258. MR 34 #7056.
8. G. T. SYMM, "Numerical mapping of exterior domains," *Numer. Math.*, v. 10, 1967, pp. 437-445. MR 36 #3525.
9. G. T. SYMM, "Conformal mapping of doubly-connected domains," *Numer. Math.*, v. 13, 1969, pp. 448-457. MR 40 #3736.

** All times given here are for the CDC 7600.